

# Arquitectura SIP de alta capacidad

---

[jesusr@voztele.com](mailto:jesusr@voztele.com)

# Necesidades generales

---

- Estabilidad
- Escalabilidad
- Redundancia
- Flexibilidad
- Interoperabilidad
- Seguridad, control de fraude y requerimientos legales
- Monitorización

# Problemas generales

---

- Interoperabilidad
  - Problemas de compatibilidad con CPEs
  - Problemas de compatibilidad con interconexiones
- NAT
  - NAT traversal en red o local... malditos ALG!!
  - Proxy de RTP o Comedia
  - Distribución geográfica

# Problemas generales

---

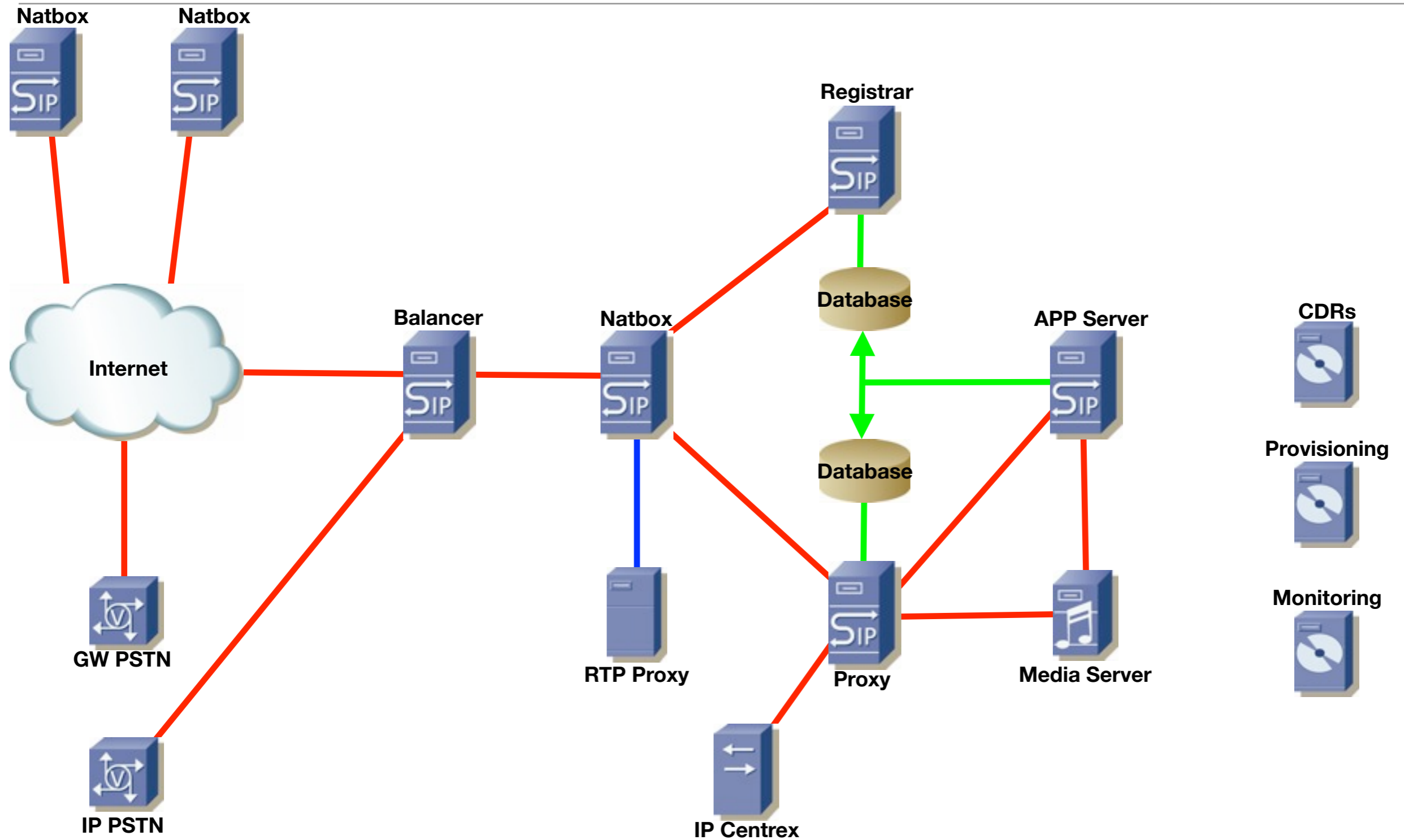
- Requerimientos legales
  - Numeración de emergencia
  - Interceptación de llamadas
- Desconocimiento de SIP
  - Fabricantes de CPE
  - Operadores para interconexiones IP
  - “Técnicos” ... out there...

# Problemas generales

---

- Calidad de servicio
  - Accesos compartidos voz y datos
    - Implementar calidad de servicio especialmente en la subida para accesos asimétricos
    - Ofrecer servicio de acceso además de la voz implementando calidad de servicio en los dos extremos
    - Uso de accesos dedicados. Voz y datos separados
- Actualizaciones, cambios, etc. Disponer de una plataforma de pruebas

# Esquema de red



# Elementos de la plataforma

---

- Proxy / Registrar / Balancer
  - SER
  - OpenSER
  - Kamailio
  - OpenSIPS
- Servidor de aplicaciones
  - WeSIP

# Elementos de la plataforma

---

- Natbox
  - Elemento que permite distribuir el NAT traversal de forma geográfica bajo el concepto de Outbound Proxy
  - Es un proxy SIP y uno (o N) proxies RTP con la única funcionalidad de modificar el SDP para mantener los RTP localmente y lo más cercano posible a los usuarios
  - Permite eliminar cualquier referencia a NAT en los proxies. Facilita mucho las configuraciones
  - Menor delay, jitter y packet loss == Mayor calidad == Cliente satisfecho

# Elementos de la plataforma

---

- Proxy RTP
  - RTPProxy
    - Escrito en C == muy rápido y eficiente
    - Balanceo de carga y tolerancia a fallos
- Base de datos
  - MySQL
    - Cluster o replicación circular

# Elementos de la plataforma

---

- Media Server
  - Asterisk... aunque teniendo en cuenta el “release engineering” actual quizás habría que explorar otras opciones
- CDRs
  - Radius
    - Radiator
    - FreeRadius

# Elementos de la plataforma

---

- Provisión
  - Oracle
- Monitorización
  - Nagios
  - Sipsack
  - Sipp
  - WeSIP

# Optimización Proxies

---

- Cargar sólo los módulos necesarios
- Aumentar en número de “childrens”
- Aumentar la memoria privada (PKG\_MEM) y compartida (SHM\_MEM)
- Desactivar listas negras automáticas
- Usar un syslog no bloqueante
- Minimizar el número de queries a las bases de datos
- Usar el módulo pike como limitador de DoS

# Optimización Proxies

---

- Configuración por bloques:
  - Carga de módulos
  - Checks iniciales: max-forwards, tamaño paquetes, respuestas keep-alives, fixups, etc
  - Usar “routes” como funciones. “Main route” llama a las funciones (dial plans internacionales, trunks, filtros, etc... volver a los apuntes de C!!
  - Optimizar el “camino” dentro del script de configuración. Primero configuraciones comunes, después configuraciones especiales

# Optimización Proxies

---

- Usar M4 para las configuraciones
- Usar Radius para los CDR
  - Posibilidad de que el servidor Radius envíe un Accounting-Response de forma inmediata (desbloqueamos los procesos del proxy)

# Optimización Proxies

---

- Reducir las queries a las imprescindibles

```
modparam("auth_db", "load_credentials",
"$avp(rpid)=rpid; $avp(nbping_type)=nbping_type; \
$avp(nbping_interval)=nbping_interval; $avp(comedia)=comedia; \
$avp(privacy)=privacy; $avp(domain)=domain; \
$avp(dialplan)=numbering_plan; $avp(billto)=ppaid_username" )
```

```
append_hf("Remote-Party-ID: <sip:$avp(rpid)@$avp(domain);user=phone>;privacy=$avp(privacy);party=calling\r\n");
append_hf("P-Asserted-Identity: <sip:$avp(rpid)@$avp(domain)>\r\n");
if (avp_check("$avp(privacy)", "eq/full/i")) {
    append_hf("Privacy: Id\r\n");
}
```

- Uso de “routes” como funciones

```
# flood detetction from same IP
if ( !route(18) && !pike_check_req()) {
    xlog("L_ALERT", "ALERT: pike block $rm from $fu (IP:$si)\n");
    exit;
};
```

# Optimización Proxies

---

- Uso de múltiples proxies RTP con balanceo de carga
  - `modparam("nathelper", "rtpproxy_sock", "udp:1.1.1.1:7898=15 udp:1.1.1.1:7899=15 udp:1.1.1.2:7898=5 udp:1.1.1.2:7899=5 udp:1.1.1.3:7898=15 udp:1.1.1.3:7899=15 udp:1.1.1.4:7898=15 udp:1.1.1.4:7899=15")`
  - RTPProxy NO es multithread por lo que es recomendable ejecutar tantos procesos como procesadores hayan en el servidor

# Ejemplos fixup

---

```
# Prevent BYE loops generated by an Asterisk bug (Asterisk answers
# a "408 Request Timeout" with an ACK and a BYE). Those BYEs do not
# have any Route header and their ruri equals the To uri.
if ( is_method("BYE") && !is_present_hf("Route") && $tu==$ru ) {
    sl_send_reply("487", "Call/leg transaction does not exist");
    xlog("L_ERR","ERROR: 408 BYE! From: $fu - To: $tu - Call-ID: $ci\n");
    exit;
}

# ACK correction for the stupid *
if ( is_method("ACK") && uri=~"sip:.*@RE(MY_DOMAIN)"
&& (search("^User-Agent:.*Asterisk") || search("^User-Agent:.*autoattendant"))
&& is_present_hf("Route")) {
    rewriteuri("sip:MY_IP;lr=on");
}

# keepalive NOTIFY and OPTIONS replied ok
if ( is_method("NOTIFY") && is_uri_host_local()
&& ($rU==NULL || search("^Event: keep-alive")) ) {
    sl_send_reply("200", "OK - keepalive");
    exit;
}
if ( is_method("OPTIONS") && is_uri_host_local() && $rU==NULL ) {
    sl_send_reply("200", "OK - keepalive");
    exit;
}
```

# Disponibilidad de la plataforma

---

- Usar sistemas en alta disponibilidad
  - Balanceadores
  - VRRP
  - Linux HA
  - Cluster MySQL

# Provisión

---

- Automatizar al máximo la provisión de usuarios y servicios
  - Provisiones manuales llevan a errores y esos errores llevan a clientes descontentos
  - Rollback. Si hay un error en algún punto de la provisión, deshacer lo que se ha provisionado. Información consistente
- Automatizar otros procesos
  - Cargas de rutas
  - Cambios de destinos y precios en sistemas prepago

# Monitorización

---

- Existen pocos sistemas especializados en SIP
- Monitorización inteligente
  - Se debe monitorizar a nivel de aplicación
  - Control del ASR medio y por destino
- Creación de scripts y aplicaciones de forma manual con aplicaciones como Sipsak, SIPp, WeSIP, Asterisk, etc
- Integración con Nagios (u otros) sistemas de gestión de alarmas

Y esto es todo... gracias  
[jesusr@voztele.com](mailto:jesusr@voztele.com)